What Is Claimed Is:

1. A processing system comprising:

a data engine adapted to identify profile data corresponding to low-level instances of a software application;

a model library adapted to store at least one model, the at least one model having high-level instances;

a model mapping engine adapted to at least one of query the data engine to obtain a list of the high-level instances, query the profile data, and map the profile data to the high-level instances; and

a visualization system adapted to present the profile data in terms of the high-level instances.

2. The processing system of claim 1, wherein the visualization system is at least one of a sampling-based profile visualization system and a call graph profile visualization system.

3. The processing system of claim 2, wherein the profile data is sampling-based profile data and the sampling-based profile visualization system is adapted to present the sampling-based profile data via an architecture view.

4. The processing system of claim 2, wherein the profile data is call graph profile data and the call graph profile visualization system is adapted to present the call graph profile data via a hierarchical view.

5. The processing system of claim 1, further comprising:

an expert system adapted to provide high-level advice relating to the low-level instances of the software application.

6. The processing system of claim 1, further comprising:

a model library browser adapted to at least one of create, edit, automatically generate, and select the at least one model.

7. The processing system of claim 6, wherein the model library browser includes at least one of a model editor adapted to edit the at least one model, and a model generator adapted to generate the at least one model.

8. The processing system of claim 1, wherein the model mapping engine is adapted to perform at least one of a top-level instance query, a high-level instances structure query, a high-level instance flattening query, and a profile data query.

9. A method comprising:

mapping profile data of a software application to low-level instances of the software application;

performing at least one of generating and selecting at least one model appropriate for the software application, the at least one model having high-level abstractions;

applying the at least one model to the profile data to map the low-level instances to the high-level abstractions; and

creating visualizations of the high-level abstractions.

10. The method of claim 9, further comprising:

providing advice to improve performance of the software application in terms of the high-level abstractions.

11. The method of claim 9, wherein said performing at least one of generating and selecting comprises at least one of creating a new model, editing an existing model, and automatically generating a model.

12. A method comprising:

collecting profile data of a software application;

selecting at least one model to analyze the profile data, the at least one model having top-level instances;

retrieving the top-level instances;

creating root node for each top level instance;

generating a hierarchical model for each root node, the hierarchical model having a plurality of child node

associating the profile data with the plurality of child nodes;

displaying the hierarchical models.

13. The method of claim 12, wherein the generating is done recursively.

14. The method of claim 12, further comprising:

traversing each hierarchical model to obtain a list of functions within the software application; and

creating a child node for each function.

15. The method of claim 12, wherein the profile data is sampling-based profile data.

16. The method of claim 12,wherein the profile data is call graph profile data.

17. A machine accessible medium containing program instructions that, when executed by a processor, cause the processor to:

map profile data of a software application to low-level instances of the software application;

at least one of generate and select at least one model appropriate for the software application, the at least one model having high-level abstractions;

apply the at least one model to the profile data to map the low-level instances to the high-level abstractions; and

create visualizations of the high-level abstractions.

18. The machine accessible medium according to claim 17, containing further program instructions that, when executed by a processor, cause the processor to:

provide advice to improve performance of the software application in terms of the high-level abstractions.

19. The machine accessible medium according to claim 17, containing further program instructions that, when executed by a processor, cause the processor to:

at least one of create a new model, edit an existing model, and automatically generate a model.

20. A machine accessible medium containing program instructions that, when executed by a processor, cause the processor to:

collect profile data of a software application;

select at least one model to analyze the profile data, the at least one model having top-level instances;

retrieve the top-level instances;

create root node for each top level instance;

generate a hierarchical model for each root node, the hierarchical model having a plurality of child node

associate the profile data with the plurality of child nodes;

display the hierarchical models.

21. The machine accessible medium according to claim 20, containing further

program instructions that, when executed by a processor, cause the processor to:

generate the hierarchical model for each node recursively.

22. The machine accessible medium according to claim 20, wherein the computer readable memory contains further program instructions that, when executed by a processor, cause the processor to:

traverse each hierarchical model to obtain a list of functions within the software application; and

create a child node for each function.

23. The machine accessible medium according to claim 20, wherein the profile data is sampling-based profile data.

24. The machine accessible medium according to claim 20, wherein the profile data is call graph profile data.